

04/26/00

UTILITY PATENT APPLICATION TRANSMITTAL <small>Only for new nonprovisional applications under 37 CFR 1.53(b))</small>		Attorney Docket No. 862.C1901 First Named Inventor or Application Identifier MIYUKI ENOKIDA ET AL. Express Mail Label No.	
APPLICATION ELEMENTS <small>See MPEP chapter 600 concerning utility patent application contents.</small>		ADDRESS TO: Assistant Commissioner for Patents Box Patent Application Washington, DC 20231	
1. <input checked="" type="checkbox"/> Fee Transmittal Form <small>(Submit an original, and a duplicate for fee processing)</small>		6. <input type="checkbox"/> Microfiche Computer Program (<i>Appendix</i>)	
2. <input checked="" type="checkbox"/> Specification Total Pages 48		7. Nucleotide and/or Amino Acid Sequence Submission <small>(if applicable, all necessary)</small>	
3. <input checked="" type="checkbox"/> Drawing(s) (35 USC 113) Total Sheets 17		a. <input type="checkbox"/> Computer Readable Copy	
4. <input checked="" type="checkbox"/> Oath or Declaration Total Pages 2		b. <input type="checkbox"/> Paper Copy (identical to computer copy)	
a. <input checked="" type="checkbox"/> Newly executed (original or copy) b. <input type="checkbox"/> Unexecuted for information purposes c. <input type="checkbox"/> Copy from a prior application (37 CFR 1.63(d)) <small>(for continuation/divisional with Box 17 completed)</small> <small>[Note Box 5 below]</small>		c. <input type="checkbox"/> Statement verifying identity of above copies	
5. <input type="checkbox"/> Incorporation By Reference (<i>useable if Box 4c is checked</i>) <small>The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4c, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.</small>		ACCOMPANYING APPLICATION PARTS	
		8. <input checked="" type="checkbox"/> Assignment Papers (cover sheet & document(s))	
		9. <input type="checkbox"/> 37 CFR 3.73(b) Statement <small>(when there is an assignee)</small> <input type="checkbox"/> Power of Attorney	
		10. <input type="checkbox"/> English Translation Document (<i>if applicable</i>)	
		11. <input type="checkbox"/> Information Disclosure Statement (IDS)/PTO-1449 <input type="checkbox"/> Copies of IDS Citations	
		12. <input type="checkbox"/> Preliminary Amendment	
		13. <input checked="" type="checkbox"/> Return Receipt Postcard (MPEP 503) <small>(Should be specifically itemized)</small>	
		14. <input type="checkbox"/> Small Entity Statement(s) <input type="checkbox"/> Statement filed in prior application <small>Status still proper and desired</small>	
		15. <input type="checkbox"/> Certified Copy of Priority Document(s) <small>(If foreign priority is claimed)</small>	
		16. <input type="checkbox"/> Other: _____	
17. If a CONTINUING APPLICATION, check appropriate box and supply the requisite information: <input type="checkbox"/> Continuation <input type="checkbox"/> Divisional <input type="checkbox"/> Continuation-in-part (CIP) of prior application No. ____/_____			
18. CORRESPONDENCE ADDRESS			
<input checked="" type="checkbox"/> Customer Number or Bar Code Label		05514 <small>(Insert Customer No. or Attach bar code label here)</small>	
NAME		<input type="checkbox"/> Correspondence address below	
Address			
City		State	Zip Code
Country		Telephone	Fax

CLAIMS	(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) CALCULATIONS
	TOTAL CLAIMS (37 CFR 1.16(c))	64-20 =	44	X \$ 18.00 =	\$792.00
	INDEPENDENT CLAIMS (37 CFR 1.16(b))	12-3 =	9	X \$ 78.00 =	\$702.00
	MULTIPLE DEPENDENT CLAIMS (if applicable) (37 CFR 1.16(d))			\$260.00 =	\$0
				BASIC FEE (37 CFR 1.16(a))	\$690.00
				Total of above Calculations =	\$2184.00
	Reduction by 50% for filing by small entity (Note 37 CFR 1.9, 1.27, 1.28).				0
				TOTAL =	\$2184.00

19. Small entity status

- a. A Small entity statement is enclosed
- b. A small entity statement was filed in the prior nonprovisional application and such status is still proper and desired.
- c. Is no longer claimed.

20. A check in the amount of \$ 2184.00 to cover the filing fee is enclosed.21. A check in the amount of \$ 40.00 to cover the recordal fee is enclosed.

22. The Commissioner is hereby authorized to credit overpayments or charge the following fees to Deposit Account No. 06-1205:

- a. Fees required under 37 CFR 1.16.
- b. Fees required under 37 CFR 1.17.
- c. Fees required under 37 CFR 1.18.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED	
NAME	LEONARD P. DIANA, Reg. 29,296
SIGNATURE	
DATE	April 25, 2000

TITLE OF THE INVENTION

DATA PROCESSING METHOD AND DATA PROCESSING DEVICE

5

FIELD OF THE INVENTION

The present invention relates to a data processing method and data processing device for registering meta-data in and distinguishing meta-data from data.

10

BACKGROUND OF THE INVENTION

Meta-data is "data about data," and is used as data which explains binary data such as video data or audio data. However, when binary data and meta-data corresponding thereto exist in separate files, when moving or copying files, 15 the user must simultaneously manage the binary data and the meta-data, which is very troublesome.

20

For this reason, in order to simplify management of binary data and meta-data, various methods of writing binary data and meta-data have conventionally been proposed. This type of conventional art can be divided into methods which stipulate a new binary format, and methods of managing in a database.

25

First, examples of methods which stipulate a new binary format include, for image data, Tiff, Exif, and Flashpix. Fig. 16 schematically shows a format which incorporates meta-data into binary data. Binary data is, for example,

image data. A typical method of incorporating meta-data into binary data, as shown in Fig. 16, is to provide in the header portion of the image data a domain for writing meta-data, into which the user writes the meta-data. Writing meta-data
5 in this way facilitates data searching and classification. Further, since the meta-data is included in the binary data, both can be managed as a single file, making file management comparatively easier.

Next, a method of managing binary data and meta-data
10 in a database will be explained. Fig. 17 schematically shows a method of managing binary data and meta-data in a database. Methods like that shown in Fig. 17, which use a database, etc. to manage binary data and meta-data existing as separate files, are widely used. These methods have the advantage that
15 existing binary data can be used as is in existing applications.

It is also commonly known that a directory structure
can be used in file management. Accordingly, when, for example, managing a large number of binary data files, all
20 of the binary data files are usually not placed in a single directory, but a plurality of sub-directories are created, one for each group of binary data files sharing a certain affiliation or characteristic. In this case, the meta-data for all of the binary data files in a given sub-directory
25 have an item in common. However, at present, each binary data file exists separately, and meta-data is managed by one of

the two methods discussed above.

Both methods which stipulate a new format for writing meta-data and methods which manage meta-data using a database have their respective problems.

5 First, when a new format for writing meta-data is stipulated, existing binary data must be converted into that new format, and then the meta-data written into the new format. Further, in order to use the meta-data in the new format for searching, applications compatible with the new format are
10 necessary. In other words, the drawback of this method is that writing and using the meta-data require a very large number of steps and a special environment. In addition, in order to process binary data in the new format (in the case of image data, to replay the image, for example),
15 applications compatible with the new data format are necessary, and existing applications cannot be used.

Moreover, since the method of writing the meta-data is uniquely determined by the new format, a further problem is that in order to prepare an application which uses
20 meta-data in the new format, it is necessary to create new routines for searching the meta-data. In addition, the prescribed format must be changed in order to write meta-data in a new domain.

On the other hand, when using a database to
25 simultaneously manage binary data and meta-data, meta-data cannot be recorded or used without database software. Further,

special software is also necessary to display recorded meta-data. In addition, when binary data is taken out of the database, it is not accompanied by the corresponding meta-data, and becomes binary data without meta-data.

When using a directory structure to manage binary data files, as well, the meta-data for binary data files in a given directory exist for each binary data file, and are managed as separate files. Consequently, even when a plurality of binary data files share an item in common, each binary data file is managed independently, thus increasing the data quantity of meta-data which must be managed. Further, when changing an item in common, meta-data for each binary data file changed must also be changed.

15 SUMMARY OF THE INVENTION

The present invention was created in view of the foregoing problems relating to writing and searching meta-data, and it is an object hereof to enable registration of meta-data in directory data without influencing existing applications.

Another object of the present invention is to use a typical data writing language to write meta-data, thus enabling the use of existing tools compatible with typical data writing languages, and facilitating the development of compatible applications.

A further object of the present invention is to enable

meta-data written in directory data to be extracted therefrom and provided for processing such as searching, reference, and modification.

A further object of the present invention is to
5 register meta-data in a directory, thereby contributing to simplification of management of meta-data and reduction of data quantity thereof.

A further object of the present invention is to enable simple attachment of meta-data to data files grouped by a
10 certain affiliation or characteristic and managed by a directory.

In order to attain the foregoing objects, in one embodiment of the present invention, meta-data is registered in directory data. By registering meta-data in directory data,
15 it becomes unnecessary to attach meta-data to each of the files under that directory, and change of the contents thereof is also simplified.

A further object of the present invention is to prevent meta-data registered in a directory from being lost or wasted
20 when a file is moved or copied from that directory.

A further object of the present invention is to enable suitable meta-data to be attached to a directory even when meta-data is already registered in each binary data.

A further object of the present invention is to
25 expedite searching by extracting suitable meta-data from a data file with meta-data registered therein, and registering

00924-355560

that meta-data in directory data.

A further object of the present invention is to simply group data files in directories by certain affiliations or characteristics.

5 Other features and advantages of the present invention will be apparent from the following description made in conjunction with the accompanying drawings, in which like reference symbols designate the same or similar members throughout the figures thereof.

10

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated in and constitute a part of the Specification, illustrate embodiments of the present invention and, together with the 15 description, serve to explain the principles of the invention.

Fig. 1 is a block diagram showing the structure of a data processing device according to a first embodiment of the present invention.

20 Fig. 2 is a flow chart explaining meta-data registration processing according to the first embodiment.

Fig. 3 is a drawing explaining registration of meta-data in directory data according to the first embodiment.

25 Fig. 4 is a flow chart showing steps for distinguishing and extracting registered meta-data according to a second

embodiment of the present invention.

Fig. 5 is a flow chart explaining the details of processing for distinguishing meta-data according to the second embodiment.

5 Fig. 6 is a drawing showing an example of data structure of directory data with XML data registered therein as meta-data.

Fig. 7 is a drawing explaining an example of a directory structure according to the second embodiment.

10 Fig. 8 is a flow chart explaining processing for revising directory data according to the second embodiment.

Fig. 9 is a flow chart showing a method of registering meta-data in binary data according to a third embodiment of the present invention.

15 Fig. 10 is a flow chart explaining a method of determining whether meta-data is registered in binary data according to the third embodiment.

Fig. 11 is a drawing showing an example of data structure of binary data with meta-data registered therein according to the third embodiment.

20 Fig. 12 is a drawing explaining an example of a method for comparing and synthesizing meta-data and directory data according to the third embodiment.

Fig. 13 is a flow chart showing a method of registering automatically generated directory meta-data in directory data according to a fourth embodiment of the present

invention.

Fig. 14 is a flow chart showing a method of automatically generating directory meta-data according to the fourth embodiment.

5 Fig. 15 is a drawing showing an example of automatic generation of directory meta-data from a list of meta-data according to the fourth embodiment.

Fig. 16 is a drawing schematically showing a format which incorporates meta-data into binary data.

10 Fig. 17 is a drawing schematically showing a method for managing binary data and meta-data in a database.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Preferred embodiments of the present invention will 15 now be described in detail with reference to the accompanying drawings.

[First Embodiment]

Fig. 1 is a block diagram showing the structure of a data processing device according to the first embodiment of 20 the present invention. In Fig. 1, 100 is a reading section, which reads an image using a scanner, for example. 101 is an input section, which inputs instructions from the user, data, etc., and which includes a keyboard, a pointing device, etc. 102 is a storage section, which stores binary data, 25 meta-data, etc. in a directory structure. A hard disk is typically used as the storage section 102. 103 is a display

section, which displays binary data accumulated in the storage section 102, and displays image data read by the reading section 100. A CRT, liquid crystal display device, etc. is typically used as the display section 103.

5 104 is a CPU, which is involved in all processing of the respective structures above, and a ROM 105 and a RAM 106 provide to the CPU 104 programs, data, and/or operating domains necessary in this processing. Here, a control program for realizing the processing steps of the present embodiment, 10 to be explained below with reference to the flow chart in Fig. 2, is also stored in the ROM 105. A structure in which this control program is stored in the storage section 102 and loaded into the RAM 106 for execution by the CPU 104 is also possible.

15 Incidentally, a data processing device according to the present embodiment is provided with various structural elements in addition to those discussed above, but these are not the principal object of the present invention, and accordingly explanation thereof will be omitted here.

20 The following will explain processing for registering meta-data in directory data using a data processing device structured as above. Fig. 2 is a flow chart explaining meta-data registration processing according to the present embodiment.

25 In Fig. 2, first, in step S301, directory data of a directory indicated by the user is read into memory (into

000240 05935560

the RAM 106). This is accomplished by, for example, inputting the name of a desired directory through the keyboard, or indicating the directory using a pointing device (such as a mouse). Next, in step S302, an XML file which records 5 meta-data, indicated by the user, is read into memory (into the RAM 106). The XML file is also indicated by inputting the file name through the keyboard, indicating a corresponding icon using a pointing device (such as a mouse), etc.

10 Next, in step S303, it is investigated whether the XML file recording meta-data is XML data of the proper format. In making this determination of proper format, it is checked whether the format for writing an XML file is satisfied (for example, whether or not the brackets at the left and right 15 of each tag form pairs, whether the form of tagging is correct, etc.). This determination of whether or not the XML data has the proper format may also include a check of whether it is valid XML data. Here, this determination of whether it is valid XML data may be accomplished by, for example, checking 20 whether or not the XML data is written in accordance with a schemer such as DTD (Document Type Definition).

If it is determined in step S303 that the XML data does not have the proper format, processing proceeds to step S307. In step S307, a message to the effect that there was an error 25 in the XML data is displayed on the display section 103, and processing ends.

09585955
09242100

If, on the other hand, it was determined in step S303 that the XML file was XML data in the proper format, processing proceeds to step S304. In step S304, the meta-data is appended to the end of the directory data read into memory in step 5 S301, thereby registering the meta-data. Then, in step S306, the directory data registering the meta-data is outputted, and processing ends. By means of the output in step S306, directory data with meta-data attached thereto, having the data structure shown in Fig. 3, is stored in the storage 10 section 102 as a single directory data file.

Fig. 3 is a drawing explaining registering of meta-data in directory data according to the present embodiment. As shown in Fig. 3, meta-data written in XML data format is appended to the end of the directory data (in the present 15 example, after the identifier <EOF> indicating the end of the directory data). In this way, the meta-data can be registered in the directory data without influencing other applications. Specifically, in referring to directory data, typical applications only use data from the beginning of the 20 directory data to the final identifier, and thus the meta-data appended thereafter does not affect the application.

Further, since the meta-data is written in XML format, by extracting this XML data portion, it is possible to add 25 to, change, or refer to the meta-data using tools which understand XML data. Thus the present invention is superior

in wide applicability. Incidentally, extraction of an XML data portion will be explained in detail in the second embodiment below.

As discussed above, in the present embodiment, by writing meta-data in XML format and appending this XML data to the end of directory data, meta-data can be registered in existing directory data, without influencing existing applications.

[Second Embodiment]

10 The first embodiment above explained a method of registering meta-data in directory data to create a directory data file. The present second embodiment will explain processing for distinguishing whether or not meta-data is registered in a directory data file, and when it is registered, 15 for extracting the meta-data. Here, a data processing device according to the present second embodiment is structured as that of the first embodiment above, and accordingly explanation thereof will be omitted.

The following will explain determination of whether 20 meta-data registered in the manner explained in the first embodiment above is present in the directory data file of an indicated directory, and operations for extracting the registered meta-data. Fig. 4 is a flow chart showing steps for distinguishing and extracting registered meta-data 25 according to the present second embodiment. Incidentally, in the present embodiment, the extracted meta-data is

00924100 - 00925560

displayed on the display section 103, but the form of output is not limited to this. It will be evident to one skilled in the art that other structures are also possible, such as one which provides the extracted meta-data for editing or

5 search processing.

As shown in Fig. 4, first, in step S501, the user indicates a directory data file for which it is to be determined whether or not meta-data is registered therein, i.e., data for processing. In step S501, indication of the data for

10 processing is accomplished by, for example, inputting the file name of a directory data file through the keyboard, or indicating a corresponding icon using a pointing device (such as a mouse).

Next, in step S502, it is determined whether or not

15 meta-data written in the XML format is registered in the data of the indicated directory data file. Here, the details of the determination processing in step S502 will be explained with reference to the flow chart in Fig. 5 and the explanatory drawing in Fig. 6. Fig. 5 is a flow chart explaining the

20 details of processing for distinguishing meta-data according to the present embodiment. Further, Fig. 6 is a drawing showing an example of data structure of directory data which registers XML data as meta-data.

As was explained in the first embodiment above, the

25 data structure of a directory data file (data for processing) with XML data as meta-data registered therein, is as shown

in Fig. 6. Accordingly, distinguishing the presence or absence of meta-data is performed as follows.

As shown in Fig. 5, first, in step S601, the entirety of the data of the directory data file (the entirety of the 5 data for processing) indicated in step S501 is read into memory (into the RAM 106). Since the data outputted in step S306 in the first embodiment above is managed as a single file, the entirety of the data can be read out using a file managing system.

10 Next, in step S602, it is checked whether the character string "</PhotoXML>" is present at the end of the data read in step S601. If this character string is not present, processing proceeds to step S605.

If the character string "</PhotoXML>" was present at the end 15 of the data for processing, processing proceeds to step S603, where it is determined whether the character string "</PhotoXML>" is preceded by a character string "<PhotoXML>". If in step S603 the character string "<PhotoXML>" was found, processing proceeds to step S604, where it is concluded that 20 meta-data is registered in the data for processing, and processing ends. If, on the other hand, the character string "<PhotoXML>" was not found, processing proceeds to step S605, where it is concluded that no meta-data is registered in the data for processing. In other words, if the character string 25 "</PhotoXML>" is not found in step S602, or if the character string "<PhotoXML>" is not found in step S603, processing

proceeds to step S605, where it is concluded that no meta-data is registered in the data for processing.

- Incidentally, in step S603, after finding the character string "<PhotoXML>", it is possible to confirm whether the
5 data bracketed by the two foregoing character strings is written in proper XML format. Further, in this case, it is also possible to perform determination of whether the data is valid XML data. Here, this determination of proper format and of valid data are as explained in the first embodiment
10 above (step S303).

The following explanation returns to the flow chart in Fig. 4. When it was concluded in the processing shown in the flow chart in Fig. 5 that meta-data is registered in the data for processing, processing proceeds to step S503. In
15 step S503, based on the XML data bracketed by the character strings "<PhotoXML>" and "</PhotoXML>", the meta-data is displayed on the display section 103, and processing ends. If, on the other hand, it was determined in step S502 that there is no meta-data registered in the data for processing,
20 processing ends.

As explained above, in the present embodiment, it is possible to distinguish directory data with meta-data recorded therein from normal directory data based on rules for writing XML data, and, when directory data recording
25 meta-data is distinguished, to display the meta-data.

In other words, in the present embodiment, it is

possible to distinguish directory data which registers meta-data from directory data which does not register meta-data, and to extract the registered meta-data. Accordingly, if an existing data writing language is used
5 for the meta-data, it is possible to, for example, edit, refer to, or search the meta-data using existing tools for that data writing language, thus saving the effort of developing new tools.

Next, a method of storing meta data for binary data
10 files stored by directory will be explained in concrete terms with reference to Fig. 7. For ease of explanation, assume that, as shown in Fig. 7, a root directory Directory 1 contains three sub-directories Directory 2, Directory 3, and Directory 4, each of which contains binary data files.

15 In such a case, if meta-data is registered in the directory data of each directory, common meta-data is stored for the Binary Data Files 2-1 through 2-3 in Directory 2, common meta-data is stored for the Binary Data Files 3-1 and 3-2 in Directory 3, and common meta-data is stored for the
20 Binary Data Files in Directory 4.

If the binary data files contain, say, image data recorded using a digital camera, a simple example of meta-data would be information indicating that the binary data in Directory 2 was recorded on March 10, and the binary
25 data in Directory 3 was recorded on March 11. Alternatively, it is also possible to divide into directories based on some

sort of affiliation, such as "Scenery" or "People." In any case, the items, contents, etc. stored in the meta-data can be determined by the manner in which data files are divided at the time of creating the sub-directories.

5 In the case above, meta-data is not stored in each binary data file, but in the directory data of each sub-directory. Accordingly, an application reading a binary data file will read out the meta-data attached to the directory the binary data file belongs to, and process this as meta-data
10 for the binary data file.

For example, in order to search for a binary data file by a certain item stored in the meta-data, in the present embodiment, since the meta-data attached to the directory data is effective for all of the binary data files belonging
15 to that directory, it is sufficient to merely search the directories, making it possible to perform search processing rapidly.

Incidentally, in the foregoing explanation, the data stored in the directories was binary data, but it may of course
20 be any type of data.

[Revision of Directory Data by File System]

When files are added to or removed from a directory the file system revises the directory data in accordance with this processing. However, if the directory data is revised
25 with the meta-data attached, the meta-data is destroyed. Accordingly, in order to handle directory data to which

000240-95985560

meta-data has been attached as explained in the first embodiment above, slight changes to the file system are necessary. The following will explain processing for revision of directory data by the file management system.

5 Fig. 8 is a flow chart explaining processing for revising directory data according to the present embodiment. If a file is added to the directory, or if a file in the directory is revised or deleted, the processing shown in Fig. 8 is carried out.

10 First, in step S701, the directory data file of the directory in question is read out. Then, in step S702, it is determined whether meta-data is attached to the directory data. This determination can be performed according to the steps shown in Fig. 5 discussed above. Then, if it is
15 determined that meta-data is registered in the directory data, processing proceeds to step S703.

In step S703, the detected meta-data is separated from the directory data and stored out of harm's way in a separate memory area. Next, in step S704, the directory data left after
20 separation of the meta-data is edited in the same manner as conventionally. Then, after editing of the directory data is completed, in step S705, the meta-data separated from the directory data in step S703 is re-appended to the end of the revised directory data. If on the other hand it was found
25 in step S702 that there is no meta-data registered in the directory data, processing proceeds to step S706, and the

directory data is revised straightaway.

In the foregoing processing, processing for revising the directory data in steps S704 and S706 may be equivalent to processing executed by conventional file managing systems.

5 Accordingly, it will be evident to one skilled in the art that the required conversion of conventional file management systems is not a major matter, and that with only minor changes they can be enabled to handle directory files according to the present embodiment.

10 As discussed in the first and second embodiments above, in the present invention, since meta-data is registered in directory data, meta-data can be registered and managed by directory.

Further, with the present invention, by appending
15 meta-data to the end of conventional directory data, meta-data can be registered in directory data without affecting existing applications.

Moreover, in the present invention, meta-data can be extracted from directory data recording meta-data and
20 provided for processing such as searching, reference, change, etc. Accordingly, by using a typical data writing language to write the meta-data, existing tools for that data writing language can be used, thus facilitating the development of compatible applications.

25 In addition, in the present invention, registering meta-data in a directory can facilitate meta-data, and can

help reduce data quantities. Consequently, when, for example, searching binary data for a given item included in the meta-data, only the directories need be searched, and the search can be performed rapidly. Further, to change a value
5 in a given item included in the meta-data, it is not necessary to change meta-data for each binary file; the meta-data for all binary files in a given directory can be changed by changing only the meta-data attached to the directory data. Consequently, meta-data can be changed rapidly. Moreover,
10 since meta-data common to a plurality of binary files belonging to a given directory can be managed as a single meta-data, data quantity can be reduced.

[Third Embodiment]

The first and second embodiments above discussed
15 registration of meta-data in directory data. If meta-data is registered in directory data, it becomes unnecessary to attach meta-data to each file in the directory, and it is also easier to change the contents of the meta-data. However, if files are moved or copied from the directory, the meta-data
20 registered in the directory may be lost, thus wasting the meta-data registered in the directory. In the present third embodiment, in order to resolve the foregoing problem, when binary data belonging to a directory is copied or moved outside the directory, meta-data is registered in that binary data. Here, a data processing device according to the present
25 third embodiment is equivalent to that of the first

embodiment (Fig. 1), and accordingly explanation thereof will be omitted here.

A method of registering meta-data in binary data according to the present third embodiment will now be explained with reference to the flow chart in Fig. 9. In the flow chart in Fig. 9, when binary data belonging to a directory is copied or moved outside the directory, meta-data is registered in that binary data.

First, in step S1301, a user indicates a binary data file in the directory to be copied or moved to another directory. The processing to be discussed below is performed on the copy of the binary data (not on the original binary data) when the operation is copying, and on the binary data after being moved, when the operation is moving. Next, in step S1302, it is determined whether the directory to which the indicated file belongs registers directory meta-data like that explained in the first and second embodiments above. This determination of whether or not directory meta-data is registered in the directory is performed as was explained above with reference to Figs. 5 and 6.

If the result of the determination in step S1302 is that there is directory meta-data registered in the directory, processing proceeds to step S1303, and if there is no directory meta-data, processing proceeds to step S1304. In step S1304, since there is no meta-data registered in the directory, the indicated binary data is copied (or moved).

When, on the other hand, there is directory meta-data, it is determined in step S1303 whether there is meta-data registered in the binary data indicated in step S1301. Processing for determining whether meta-data is registered
5 in the binary data will be explained with reference to Figs. 10 and 11. Fig. 10 is a flow chart showing processing for determining whether meta-data is registered in the binary data, and Fig. 11 is a drawing explaining registering of meta-data in binary data according to the present embodiment.

10 Binary data with meta-data registered therein is made up of the parts shown in Fig. 11. Accordingly, in order to distinguish the presence or absence of meta-data, in Fig. 5, first, in step S1601, the entirety of the indicated binary data is read into memory. Next, in step S1602, it is checked
15 whether the character string "</PhotoXML>" is present at the end of the binary data read into memory. If this character string is present, processing proceeds to step S1603, where it is checked whether a character string "<PhotoXML>" is present before the character string "</PhotoXML>". If this
20 character string is also present, processing proceeds to step S1604, where it is concluded that meta-data is registered in the binary data.

On the other hand, if it is judged in step S1602 that the character string "</PhotoXML>" is not present, or if it
25 is judged in step S1603 that the character string "<PhotoXML>" is not present, then processing proceeds to step

00000000000000000000000000000000

S1605, where it is concluded that no meta-data is registered in the binary data.

If it is determined by the foregoing determination processing that there is no meta-data registered in the
5 binary data, processing proceeds to step S1308. When processing has proceeded to step S1308, it means that directory meta-data is registered in the directory, but no meta-data is registered in the binary data. Accordingly, the directory meta-data is registered in the binary data, the
10 binary data is copied (or moved), and processing ends. Here, meta-data is registered in the binary data by the same method as that for registering meta-data in directory data explained in the first embodiment above. In other words, data having the form shown in Fig. 11 is generated by appending the
15 meta-data to the end of the binary data.

If, on the other hand, it was determined in step S1303 that meta-data is registered in the binary data, processing proceeds to step S1305. In step S1305, the directory meta-data registered in the directory and the meta-data registered in the binary data are compared. Then, in step
20 S1306, meta-data is prepared which, as a result of the comparison in step S1305, includes all of the directory meta-data and binary data meta-data.

Here, processing for generating the meta-data in step
25 S1306 will be explained in detail. As shown in Fig. 12, directory meta-data 1801 and meta-data 1802 are combined to

create meta-data 1803, which includes all the contents of both meta-data.

The meta-data prepared as above is then newly registered in the binary data, which is then copied (or moved),
5 and processing ends.

As discussed above, in the present embodiment, when copying or moving binary data belonging to a directory, directory meta-data registered in the directory is registered in the binary data as meta-data, and thus even
10 if some of the files in a directory are moved elsewhere, the meta-data is not lost.

Further, even when meta-data is already registered in the binary data, this meta-data can be synthesized with the directory meta-data, and thus meta-data combining the
15 directory meta-data and the meta-data can be registered in the binary data.

As is clear from the foregoing explanation of the present embodiment, in the present invention, directory meta-data written in the same data writing language as the
20 directory data is registered in the binary data, and thus it is possible to copy or move the binary data while preserving the directory meta-data information. In addition, since the meta-data is written in a data writing language, existing tools for that data writing language can be used as is, thus
25 saving the effort of developing new tools.

[Fourth Embodiment]

In cases when binary data already has meta-data, the present fourth embodiment enables registering of suitable meta-data in directory data. Here, a data processing device according to the present fourth embodiment is equivalent to

- 5 that of the first embodiment (Fig. 1), and accordingly explanation thereof will be omitted here. A method of generating suitable meta-data from binary data in a directory will now be explained with reference to the flow chart in

Fig. 13.

- 10 In Fig. 13, first, in step S2301, the user indicates a desired directory (a directory for which meta-data is to be generated and to which the meta-data is to be attached). Then, in step S2302, one of the data files in that directory is selected, and it is determined whether or not meta-data is registered
15 in that file. The steps for distinguishing the presence or absence of meta-data are as explained above with reference to Figs. 10 and 11.

- When there is meta-data registered in the binary data, processing proceeds to step S2303, where the file name of
20 the binary data registering meta-data, and the meta-data are added to a list 1 in memory. On the other hand, if in step S2302 there was no meta-data registered in the binary data, processing proceeds to step S2304. In step S2304, the file name of the binary data not registering meta-data is added
25 to a list 2 in memory. The foregoing processing is executed for all of the files in the directory (step S2305).

Next, in step S2306, directory meta-data is generated from list 1, which records the file names of binary data registering meta-data, and those meta-data. Processing for generating directory meta-data from list 1 will be explained
5 below with reference to Figs. 14 and 15. Fig. 14 is a flow chart explaining steps for generating directory meta-data in step S2306, and Fig. 15 is a drawing explaining generation of directory meta-data in step S2306.

In step S2401, it is checked whether all of the meta-data
10 recorded in list 1 share one or more meta-data item(s) in common. Here, if all of the meta-data have one or more meta-data item(s) in common, processing proceeds to step S2404, where the meta-data item(s) in common is/are extracted, directory meta-data is generated using the extracted
15 meta-data item(s), and processing ends. This processing will now be explained with reference to Fig. 15. From list 1 (501), it can be seen that meta-data is attached to the three image files img001.jpg, img002.jpg, and img003.jpg (filenames are the character strings bracketed by <Filename> and
20 </Filename>, and meta-data are the data bracketed by <PhotoXML> and </PhotoXML>). Extracting the meta-data items shared by the three data files listed in list 1 (501) and generating directory meta-data therefrom yields the directory meta-data (502).
25 If, in step S2401, there were no meta-data items shared by all of the meta-data, processing proceeds to step S2403,

where directory meta-data is generated using the meta-data item(s) shared by the most meta-data. Then, in step S2404, a list 3 is generated, which lists the names of data files registering meta-data which does not include the meta-data 5 item(s) included in the directory meta-data. Then, in step S2405, the file names listed in list 3 and their meta-data are deleted from list 1.

Then, returning to Fig. 13, in step S2307, a new directory is generated, and all of the data files recorded in list 1 10 are moved to this new directory. Incidentally, in step S2307, the new directory is given a suitable name taken from the directory meta-data. Next, in step S2308, it is determined whether list 3 has been generated; if it has been generated, processing proceeds to step S2309, but if not, step S2309 15 is skipped. In step S2309, the data files recorded in list 3 are moved to a directory named "MetaMisc." Finally, in step S2310, directory meta-data is registered in the new directory generated in step S2307, and processing ends. Here, the method of registering the meta-data in the new directory is 20 as explained in the first embodiment above, and has a data structure like that shown in Fig. 6.

The foregoing processing makes it possible to automatically generate directory meta-data from the data files in the directory, and to register this meta-data in the directory. 25 As discussed above, in the present embodiment, directory meta-data can be automatically generated from data files in

a directory, thus making it easy to register directory meta-data in the directory. Even when there are no meta-data items in common, by automatically generating directory meta-data from the binary data files in the "MetaMisc" 5 directory, new directories and directory meta-data therefor can be repeatedly generated automatically.

Incidentally, in the present embodiment, when all of the data files in the indicated directory are listed in list 1 (i.e., when all of the data files in the directory register 10 meta-data, and all of the meta-data share one or more meta-data item(s) in common), the meta-data generated may be registered in the directory data of the indicated directory, without generating a new directory.

As is clear from the foregoing explanation of the 15 fourth embodiment, the present invention makes it possible to automatically generate directory meta-data from binary data. Further, attaching meta-data to the directory enables high-speed searching. Moreover, since the meta-data is written in an existing data writing language, existing tools 20 for that data writing language can be used as is, thus saving the trouble of developing new tools.

Incidentally, in each of the foregoing embodiments, the meta-data was XML data, but there is no limitation to this. For example, it may be written in a data writing language 25 such as SGML or HTML. Further, the data of the files in each directory may be binary data such as still image data, dynamic

009240 * 5535560

image data, or audio data, or any other form of data.

The present invention may be applied to a system made up of a plurality of devices (such as host computer, interface device, reader, printer, etc.), or to a single device (such 5 as a copy machine, facsimile, etc.).

Further, the objects of the present invention may of course also be attained by supplying a system or a device with a memory medium recording the program code of software for realizing the functions of the foregoing embodiments, 10 and using a computer (or CPU or MPU) of the device or system to read out and execute the program code stored in the memory medium.

In such a case, the program code itself, read out from the memory medium, realizes the functions of the embodiments 15 discussed above, and the memory medium recording that program code structures the present invention.

The memory medium for supplying the program code may be a floppy disk, hard disk, optical disk, magneto-optical disk, CD-ROM, CD-R, magnetic tape, non-volatile memory card, 20 ROM, etc.

In addition, it goes without saying that the scope of the present invention covers not only a case in which the functions of the foregoing embodiments are realized by executing the program code read out by a computer, but also 25 a case in which, based on the instructions of the program code, an OS (operating system) operating in the computer

09538656 - 0924500

performs part or all of the actual processing, and this processing realizes the functions of the foregoing embodiments.

Furthermore, it goes without saying that the scope of
5 the present invention also covers a case in which the program code read out from the memory medium is written into the memory of an expansion board or expansion unit inserted into or connected to the computer, and, based on the instructions of the program code, a CPU, etc. of the expansion board or
10 expansion unit performs part or all of the actual processing, and this processing realizes the functions of the foregoing embodiments.

As many apparently widely different embodiments of the present invention can be made without departing from the
15 spirit and scope thereof, it is to be understood that the invention is not limited to the specific embodiments thereof except as defined in the claims.

What Is Claimed Is:

1. A data processing method comprising the steps of:
 (a) reading directory data corresponding to a
5 directory in a file management system which manages files
using a directory structure;
 (b) reading meta-data to be attached to the directory
data;
 (c) appending the meta-data read in said step (b) to
10 the end of the directory data read in said step (a); and
 (d) outputting as a single directory data file the
entirety of the data obtained in said step (c).
2. The method set forth in claim 1, further comprising
15 the step of:
 (e) determining whether or not the meta-data read in
said step (b) is written in a proper format for a predetermined
data writing language;
 wherein, in said step (c), the meta-data is appended
20 to the end of the directory data if it is determined in said
step (e) that the meta-data is written in the proper format.
3. The method set forth in claim 2, wherein:
 in said step (e), it is determined whether or not the
25 meta-data is valid as the predetermined data writing
language.

09558656.042600

4. The method set forth in claim 2, wherein:
in said step (e), it is determined whether or not the
meta-data is well-formed as the data writing language.

5

5. A method of determining whether meta-data is
registered in data of a directory data file, comprising the
steps of:

(a) reading a directory data file; and
10 (b) inspecting data of the directory data file read
in said step (a), from the end toward the beginning thereof,
and distinguishing the meta-data included in the data by
determining whether or not there is data written in a proper
format for a predetermined data writing language.

15

6. The method set forth in claim 5, further comprising
the step of:

(c) when meta-data is distinguished in said step (b),
extracting and outputting the distinguished meta-data.

20

7. The method set forth in claim 6, wherein:
in said step (c), display is performed based on the
extracted meta-data.

25

8. The method set forth in claim 6, wherein:
in said step (e), the extracted meta-data is provided

to a tool which performs predetermined processing based on the predetermined data writing language.

9. The method set forth in claim 5, wherein said step (b)

5 further comprises the steps of:

(c) checking whether or not a final character string stipulated in the predetermined data writing language is present at the end of the data; and

(d) when the final character string is present,
10 searching for an initial character string stipulated in the
predetermined data writing language, from the end toward the
beginning of the data;

wherein, when there is data between the final character string and the initial character string, this data is distinguished as meta-data.

10. The method set forth in claim 9, wherein said step (b) further comprises the step of:

(e) investigating whether or not the data bracketed by the
final character string and the initial character string has
a proper format for the predetermined data writing language.

11. The method set forth in claim 10, wherein:

the investigation in said step (e) also includes
25 determination of whether the data bracketed by the final character string and the initial character string is valid

as the predetermined data writing language.

12. The method set forth in claim 10, wherein:

the investigation in said step (e) also includes
5 determination of whether the data bracketed by the final
character string and the initial character string is
well-formed as the predetermined data writing language.

13. A data processing method comprising the steps of:

10 (a) reading directory data corresponding to a
directory to which belongs data for processing;

(b) extracting meta-data from the directory data read
in said step (a); and

(c) based on the meta-data extracted in said step (b),
15 attaching meta-data to the data for processing.

14. The method set forth in claim 13, wherein said step

(c) further comprises the step of:

20 (d) determining whether meta-data is registered in the
data for processing;

wherein, when it is determined that no meta-data is
registered in the data for processing, the meta-data
extracted in said step (b) is attached to the data for
processing.

25

15. The method set forth in claim 13, wherein said step

卷之三

(c) further comprises the steps of:

(d) determining whether meta-data is registered in the data for processing;

(e) when it is determined that meta-data is registered
5 in the data for processing, the meta-data is separated from
the data for processing; and

(f) generating new meta-data based on the meta-data separated in said step (e) and the meta-data extracted in said step (b);

10 wherein the meta-data generated in said step (f) is attached to the data for processing remaining after separation of the meta-data in said step (e).

16. The method set forth in claim 15, wherein:

15 in said step (f), new meta-data is generated so as to include all data items included in the meta-data obtained in said step (e) and the meta-data obtained in said step (b).

17. The method set forth in claim 13, wherein:
20 the data for processing is data copied to another
directory.

18. The method set forth in claim 13, wherein:
the data for processing is data moved to another
25 directory.

19. The method set forth in claim 13, wherein:
the data for processing includes a binary data portion;
and
in said step (c), the meta-data is appended after the
5 binary data portion.

20. The method set forth in claim 13, wherein:
the data for processing is image data, audio data, or
dynamic image data.

10

21. A data processing method comprising the steps of:
(a) reading data files belonging to an indicated
directory;
(b) extracting meta-data from the data files read in
15 said step (a);
(c) generating meta-data for the directory based on
the meta-data extracted in said step (b); and
(d) attaching the meta-data generated in said step (c)
to directory data.

20

22. The method set forth in claim 21, wherein:
in said step (c), the meta-data for the directory is
generated based on a meta-data item included in all of the
meta-data extracted in said step (b).

25

23. The method set forth in claim 21, wherein:

00000000000000000000000000000000

in said step (c), the meta-data for the directory is generated based on a meta-data item shared by the most meta-data extracted in said step (b).

5 24. The method set forth in claim 21, further comprising the step of:

(e) generating a new directory, and recording therein data files to which are attached meta-data which includes meta-data items used in the meta-data for the directory
10 generated in said step (c);

wherein, in said step (d), the meta-data generated in said step (c) is attached to directory data corresponding to the new directory.

15 25. The method set forth in claim 21, further comprising the step of:

(e) generating a new directory, and recording therein data files to which are attached meta-data which does not include meta-data items used in the meta-data for the
20 directory generated in said step (c).

26. The method set forth in claim 21, wherein:
the data file is an image data file, an audio data file, or
a dynamic image data file.

25

27. The method set forth in claim 26, wherein:

in said step (d), the meta-data generated in said step
(c) is appended to the end of the directory data.

28. The data processing method set forth in claim 1,
5 wherein the meta-data is written in the data writing language
XML.

29. The data processing method set forth in claim 1,
wherein the meta-data is written in the data writing language
10 SGML.

30. The data processing method set forth in claim 1,
wherein the meta-data is written in the data writing language
HTML.

15

31. A data processing device comprising:

first reading means, for reading directory data
corresponding to a directory in a file management system
which manages files using a directory structure;

20

second reading means, for reading meta-data to be
attached to the directory data;

appending means, for appending the meta-data read by
said second reading means to the end of the directory data
read by said first reading means; and

25

output means, for outputting as a single directory data
file the entirety of the data obtained by said appending

019558655 - 012600

means.

32. The device set forth in claim 31, further comprising:
determining means, for determining whether or not the
5 meta-data read by said second reading means is written in
a proper format for a predetermined data writing language;
wherein said appending means append the meta-data to
the end of the directory data if the determining means
determine that the meta-data is written in the proper format.

10

33. The device set forth in claim 32, wherein:
said determining means determine whether or not the
meta-data is valid as the predetermined data writing
language.

15

34. The device set forth in claim 32, wherein:
said determining means determine whether or not the
meta-data is well-formed as the data writing language.

20 35. A data processing device which determines whether
meta-data is registered in data of a directory data file,
comprising:
reading means, for reading a directory data file; and
distinguishing means, for inspecting data of the
25 directory data file read by said reading means, from the end
toward the beginning thereof, and distinguishing the

00000000 - 00000000

meta-data included in the data by determining whether or not there is data written in a proper format for a predetermined data writing language.

5 36. The device set forth in claim 35, further comprising:
 output means which, when said distinguishing means distinguish meta-data, extract and output the distinguished meta-data.

10 37. The device set forth in claim 36, wherein:
 said output means perform display based on the extracted meta-data.

15 38. The device set forth in claim 36, wherein:
 said output means provide the extracted meta-data to a tool which performs predetermined processing based on the predetermined data writing language.

39. The device set forth in claim 35, said distinguishing
20 means further comprising:
 checking means, for checking whether or not a final character string stipulated in the predetermined data writing language is present at the end of the data; and
 search means which, when the final character string
25 is present, search for an initial character string stipulated in the predetermined data writing language, from the end

0002400000000000

toward the beginning of the data;

wherein, when there is data between the final character string and the initial character string, said distinguishing means distinguish this data as meta-data.

5

40. The device set forth in claim 39, said distinguishing means further comprising:

investigating means, for investigating whether or not the data bracketed by the final character string and the initial
10 character string has a proper format for the predetermined data writing language.

41. The device set forth in claim 40, wherein:
said investigating means also determine whether the data
15 bracketed by the final character string and the initial character string is valid as the predetermined data writing language.

42. The device set forth in claim 40, wherein:

20 said investigating means also determine whether the data bracketed by the final character string and the initial character string is well-formed as the predetermined data writing language.

25 43. A data processing device comprising:
reading means, for reading directory data

corresponding to a directory to which belongs data for processing;

extracting means, for extracting meta-data from the directory data read by said reading means; and

5 attaching means which, based on the meta-data extracted by said extracting means, attach meta-data to the data for processing.

44. The device set forth in claim 43, said attaching means
10 further comprising:

 determining means, for determining whether meta-data is registered in the data for processing;

 wherein, when it is determined that no meta-data is registered in the data for processing, said attaching means
15 attach the extracted meta-data to the data for processing.

45. The device set forth in claim 43, said attaching means
further comprising:

 determining means, for determining whether meta-data
20 is registered in the data for processing;

 separating means which, when it is determined that meta-data is registered in the data for processing, separate the meta-data from the data for processing; and

 generating means, for generating new meta-data based
25 on the meta-data separated by said separating means and the meta-data extracted by said extracting means;

00000000-0000-0000-0000-000000000000

wherein said attaching means attach the meta-data generated by said generating means to the data for processing remaining after separation of the meta-data by said separating means.

5

46. The device set forth in claim 45, wherein:
said generating means generate the new meta-data such that it includes all data items included in the meta-data obtained by said separating means and the meta-data obtained by said extracting means.

10

47. The device set forth in claim 43, wherein:
the data for processing is data copied to another directory.

15

48. The device set forth in claim 43, wherein:
the data for processing is data moved to another directory.

20

49. The device set forth in claim 43, wherein:
the data for processing includes a binary data portion;
and
said attaching means append the meta-data after the binary data portion.

25

50. The device set forth in claim 43, wherein:

the data for processing is image data, audio data, or dynamic image data.

51. A data processing device comprising:

5 reading means, for reading data files belonging to an indicated directory;

extracting means, for extracting meta-data from the data files read by said reading means;

10 generating means, for generating meta-data for the directory based on the meta-data extracted by said extracting means; and

attaching means, for attaching the meta-data generated by said generating means to directory data.

15 52. The device set forth in claim 51, wherein:

said generating means generate the meta-data for the directory based on a meta-data item included in all of the meta-data extracted by said extracting means.

20 53. The device set forth in claim 51, wherein:

said generating means generate the meta-data for the directory based on a meta-data item shared by the most meta-data extracted by said extracting means.

25 54. The device set forth in claim 51, further comprising:
first recording means, for generating a new directory,

00000000000000000000000000000000

and recording therein data files to which are attached meta-data which includes meta-data items used in the meta-data for the directory generated by said generating means;

5 wherein said attaching means attach the meta-data generated by said generating means to directory data corresponding to the new directory.

10 55. The device set forth in claim 51, further comprising:
 second recording means, for generating a new directory,
and recording therein data files to which are attached meta-data which does not include meta-data items used in the meta-data for the directory generated by said generating means.

15 56. The device set forth in claim 51, wherein:
 the data file is an image data file, an audio data file, or
 a dynamic image data file.

20 57. The device set forth in claim 51, wherein:
 said attaching means append the meta-data generated by said generating means to the end of the directory data.

25 58. The device set forth in claim 31, wherein the meta-data is written in the data writing language XML.

59. The device set forth in claim 31, wherein the meta-data
is written in the data writing language SGML.

60. The device set forth in claim 31, wherein the meta-data
5 is written in the data writing language HTML.

61. A memory medium storing a control program to be
executed by a computer, said control program comprising code
for performing the steps of:

10 (a) reading directory data corresponding to a
directory in a file management system which manages files
using a directory structure;

(b) reading meta-data to be attached to the directory
data;

15 (c) appending the meta-data read in said step (b) to
the end of the directory data read in said step (a); and

(d) outputting as a single directory data file the
entirety of the data obtained in said step (c).

20 62. A memory medium storing a control program to be
executed by a computer, said control program comprising code
for performing the steps of:

(a) reading a directory data file; and

25 (b) inspecting data of the directory data file read
in said step (a), from the end toward the beginning thereof,
and distinguishing the meta-data included in the data by

determining whether or not there is data written in a proper format for a predetermined data writing language.

63. A memory medium storing a control program to be
5 executed by a computer, said control program comprising code for performing the steps of:

- (a) reading directory data corresponding to a directory to which belongs data for processing;
- 10 (b) extracting meta-data from the directory data read in said step (a); and
- (c) based on the meta-data extracted in said step (b), attaching meta-data to the data for processing.

64. A memory medium storing a control program to be
15 executed by a computer, said control program comprising code for performing the steps of:

- (a) reading data files belonging to an indicated directory;
- 20 (b) extracting meta-data from the data files read in said step (a);
- (c) generating meta-data for the directory based on the meta-data extracted in said step (b); and
- (d) attaching the meta-data generated in said step (c) to directory data.

ABSTRACT OF THE DISCLOSURE

A data processing device first reads directory data provided for each directory by a file management system which
5 manages files using a directory structure. When meta-data to be attached to the directory data is written in proper XML format, the device appends the meta-data to the end of the read directory data, and outputs the entirety of the data obtained thereby as a directory data file. By appending the
10 meta-data to the end of the directory data in this way, meta-data can be registered by directory with respect to existing data managed by the directory, without affecting existing applications.

15

009240 · 95985560

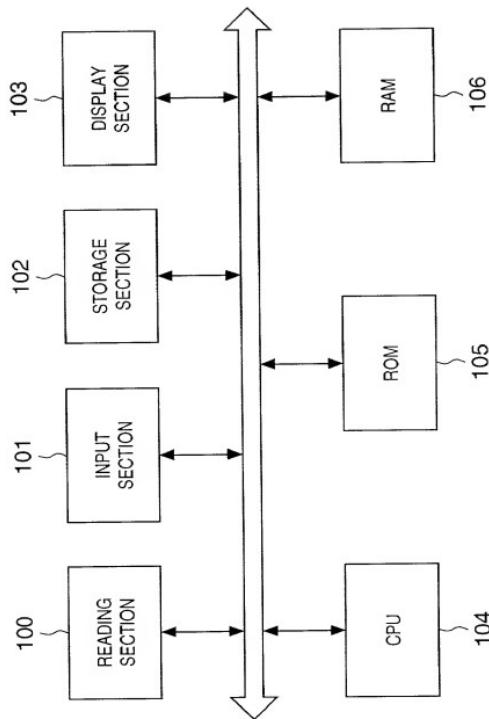
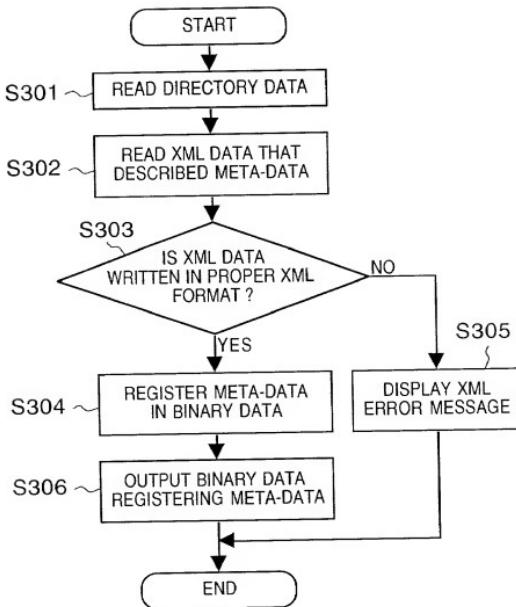
FIG. 1

FIG. 2



09558655 - 042600

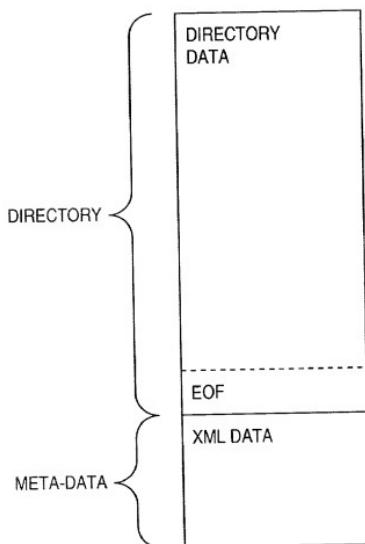
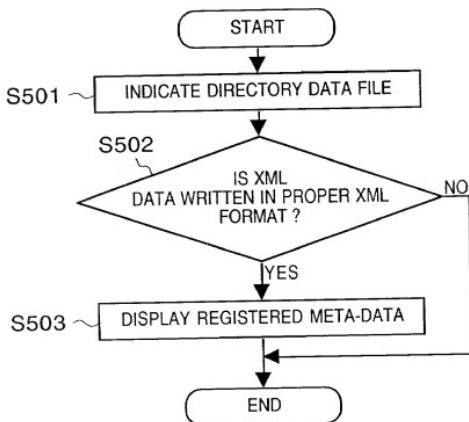
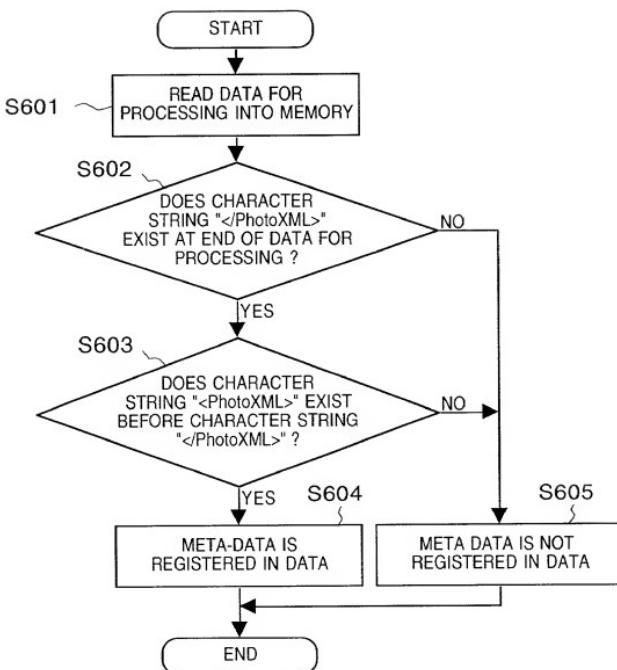
FIG. 3

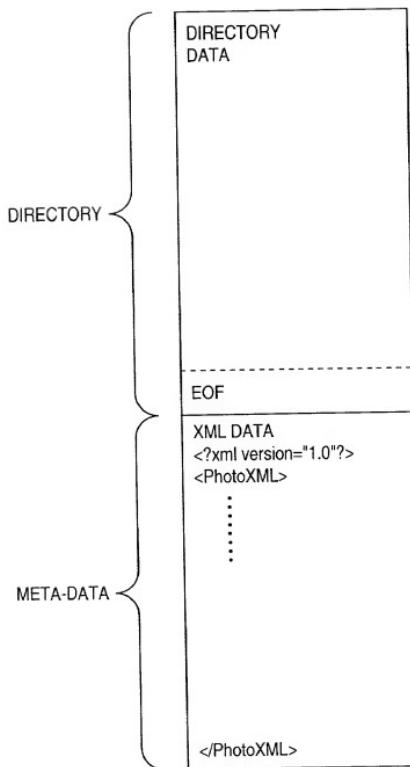
FIG. 4

095558656 - 042600

FIG. 5



0009240-042600

F I G. 6

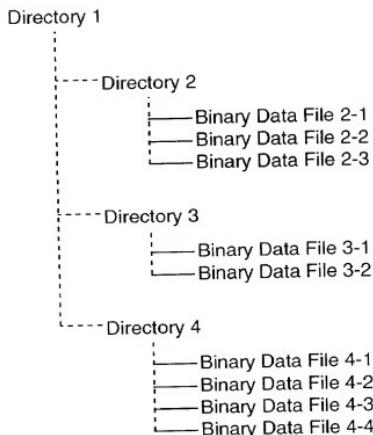
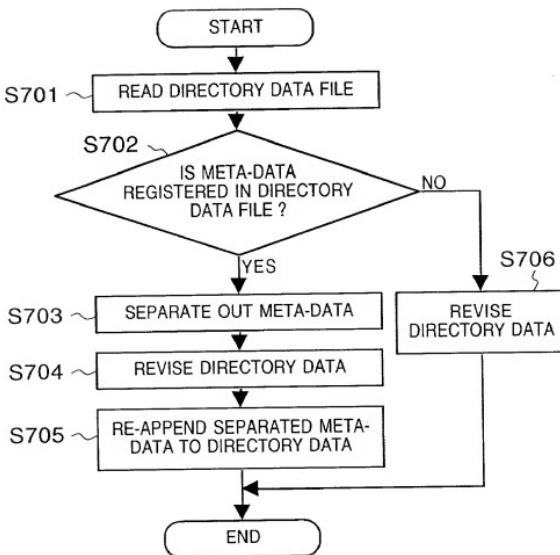
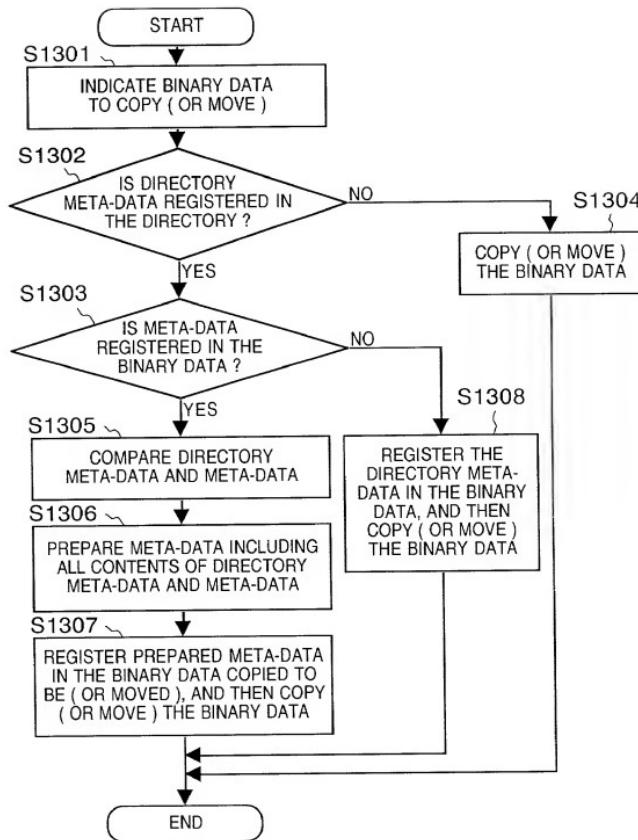
F I G. 7

FIG. 8



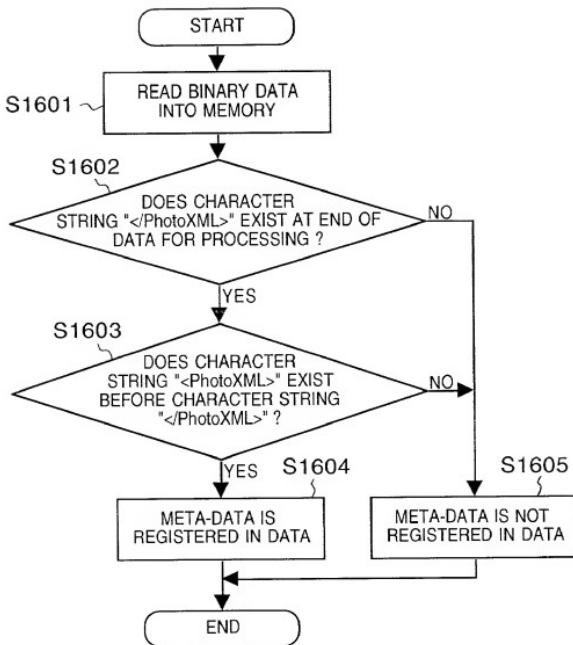
09558656 - 042600

FIG. 9



09558556 - 042600

FIG. 10



0005558655 - 042600

FIG. 11

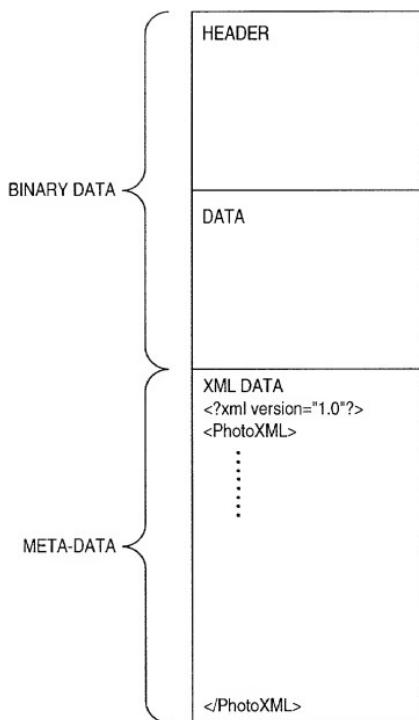


FIG. 12

096290 - 959885560

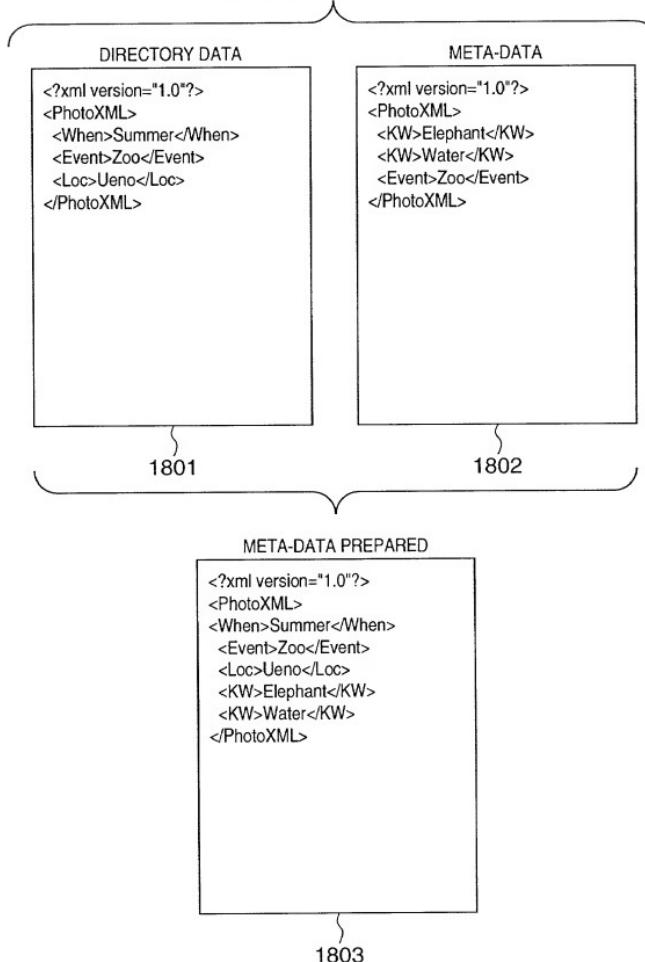
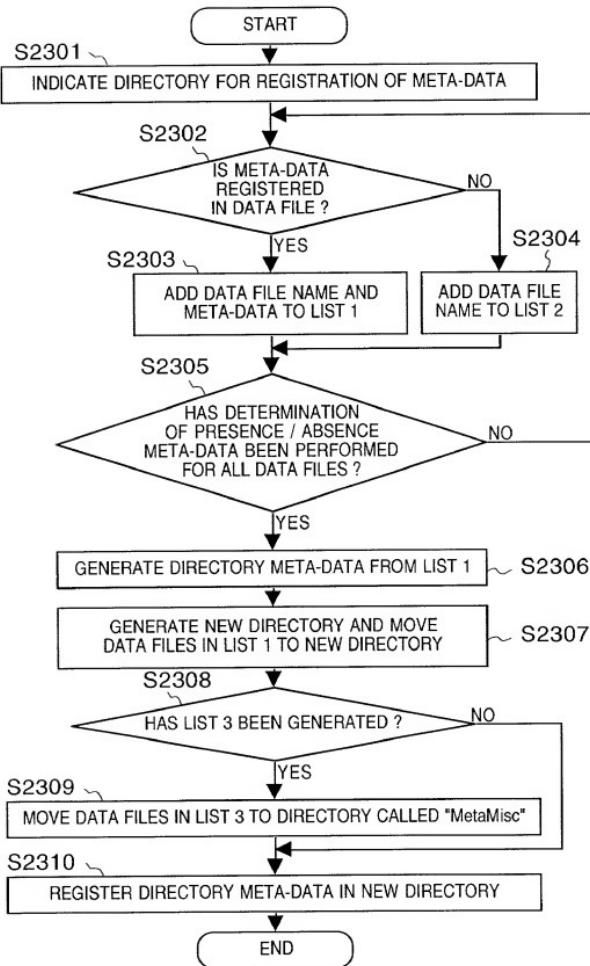
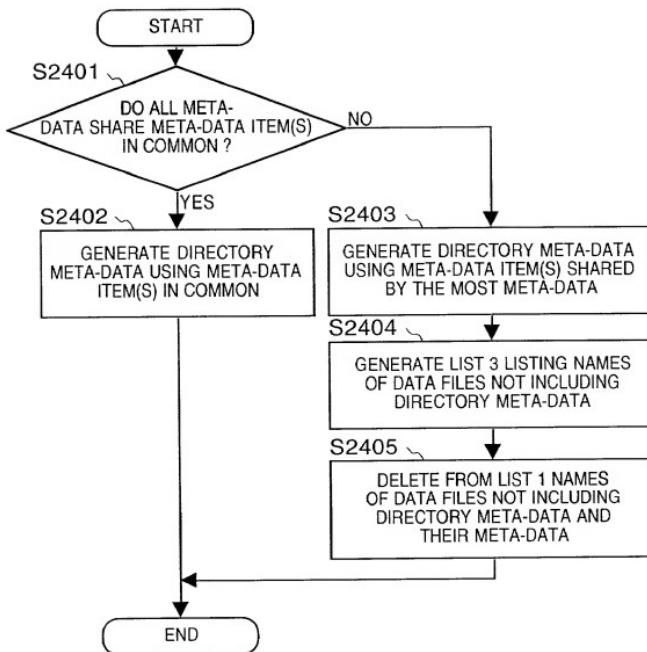


FIG. 13



095588656 - 042600

FIG. 14



0092410 - 95985560

FIG. 15

LIST 1

```

<Filename>img001.jpg</Filename>
<PhotoXML>
<When>Summer</When>
<Event>Zoo</Event>
<Loc>Ueno</Loc>
<KW>elephant</KW>
</PhotoXML>
<Filename>img002.jpg</Filename>
<PhotoXML>
<When>Summer</When>
<Event>Zoo</Event>
<Loc>Ueno</Loc>
<KW>giraffe</KW>
</PhotoXML>
<Filename>img003.jpg</Filename>
<PhotoXML>
<When>Summer</When>
<Event>Zoo</Event>
<Loc>Ueno</Loc>
<KW>monkey</KW>
</PhotoXML>

```

DIRECTORY META
GENERATED

```

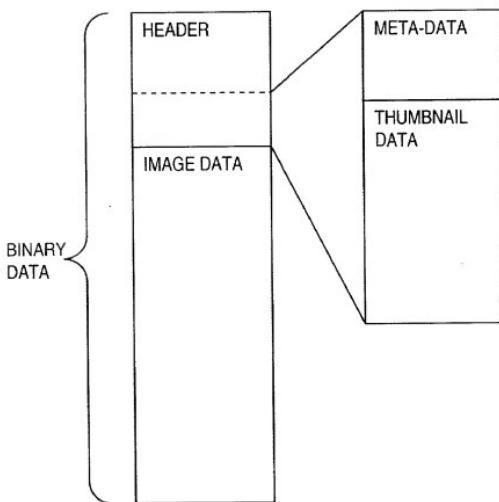
<PhotoXML>
<When>Summer</When>
<Event>Zoo</Event>
<Loc>Ueno</Loc>
</PhotoXML>

```

502

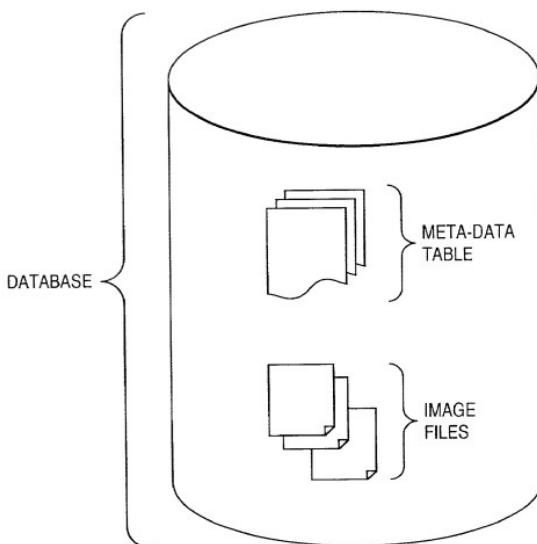
501

F I G. 16



09558656 - 042600

FIG. 17



09558656 - 042600

LFM 1901 US
P200-0094US

**COMBINED DECLARATION AND POWER OF ATTORNEY
FOR PATENT APPLICATION**
(Page 1)

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name;

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

DATA PROCESSING METHOD AND DATA PROCESSING DEVICE

the specification of which [X]is attached hereto. [] was filed on _____

as United States Application No. or PCT International Application No. _____
and was amended on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR §1.56.

I hereby claim foreign priority benefits under 35 U.S.C. §119(a)-(d) or §365(b), of any foreign application(s) for patent or inventor's certificate, or §365(a) of any PCT international application which designates at least one country other than the United States, listed below and have also identified below any foreign application for patent or inventor's certificate, or PCT international application having a filing date before that of the application on which priority is claimed:

<u>Country</u>	<u>Application No.</u>	<u>Filed (Day/Mo/Yr.)</u>	(Yes/No) <u>Priority Claimed</u>
JAPAN	11-120713	27/04/1999	Yes
JAPAN	2000-109923	11/04/2000	Yes

I hereby appoint the practitioners associated with the firm and customer number provided below to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith, and direct that all correspondence be addressed to the address associated with that Customer Number:

**FITZPATRICK, CELLA, HARPER & SCINTO
Customer Number: 05514**

COMBINED DECLARATION AND POWER OF ATTORNEY
FOR PATENT APPLICATION
(Page 2)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole or First Inventor Miyuki ENOKIDA

Inventor's signature Miyuki Enokida

Date April 19, 2000 Citizen/Subject of JAPAN

Residence 198-1-103, Sanmaicho, Kanagawa-ku, Yokohama-shi,
Kanagawa-ken, Japan

Post Office Address c/o CANON KABUSHIKI KAISHA,
30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, Japan

Full Name of Second Joint Inventor, if any Kiyoshi KUSAMA

Second Inventor's signature Kiyoshi Kusama

Date April 19, 2000 Citizen/Subject of JAPAN

Residence 12-28-202, Miyauchi 4-chome, Nakahara-ku,
Kawasaki-shi, Kanagawa-ken, Japan

Post Office Address c/o CANON KABUSHIKI KAISHA,
30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, Japan

Full Name of Third Joint Inventor, if any _____

Third Inventor's signature : _____

Date _____ Citizen/Subject of _____

Residence _____

Post Office Address _____

Full Name of Fourth Joint Inventor, if any _____

Fourth Inventor's signature _____

Date _____ Citizen/Subject of _____

Residence _____

Post Office Address _____

Full Name of Fifth Joint Inventor, if any _____

Fifth Inventor's signature _____

Date _____ Citizen/Subject of _____

Residence _____

Post Office Address _____